```
                                   H 5




FFFFFFFFFFFFFFF         111              111        XXX         XXX
FFFFFFFFFFFFFFF         111              111        XXX         XXX
FFFFFFFFFFFFFFF         111              111        XXX         XXX
FFF                  111111           111111        XXX         XXX
FFF                  111111           111111        XXX         XXX
FFF                  111111           111111        XXX         XXX
FFF                     111              111           XXX   XXX
FFF                     111              111           XXX   XXX
FFF                     111              111           XXX   XXX
FFFFFFFFFFFF            111              111              XXX
FFFFFFFFFFFF            111              111              XXX
FFFFFFFFFFFF            111              111              XXX
FFF                     111              111           XXX   XXX
FFF                     111              111           XXX   XXX
FFF                     111              111           XXX   XXX
FFF                     111              111        XXX         XXX
FFF                     111              111        XXX         XXX
FFF                     111              111        XXX         XXX
FFF                     111              111        XXX         XXX
FFF                     111              111        XXX         XXX
FFF                     111              111        XXX         XXX
FFF                  111111111        111111111     XXX         XXX
FFF                  111111111        111111111     XXX         XXX
FFF                  111111111        111111111     XXX         XXX
```

```
_$25

Symb
----
IOC$
IO_
IO_
IO_
IO_
IO_
KILL

KILL
KILL
LB_
LB_
LB_
LB_
LB_
LOCA
LOCA
LOCK

LOCK
LOCK
LOCK
LOC_
LOC_
L_CO
L_CO
L_DA
L_DA
MAIN
MAKE
MAKE
MAKE
MAKE
MAKE
MAKE

MAKE
MAKE
MAP_
MAP_

MAP
MARK
MARK
MARK
MARK
MARK
```

```
**FILE**ID**DIRACC                                F 11




DDDDDDDD      IIIIII    RRRRRRRR      AAAAAA      CCCCCCCC    CCCCCCCC
DDDDDDDD      IIIIII    RRRRRRRR      AAAAAA      CCCCCCCC    CCCCCCCC
DD      DD      II      RR      RR  AA      AA  CC          CC
DD      DD      II      RR      RR  AA      AA  CC          CC
DD      DD      II      RR      RR  AA      AA  CC          CC
DD      DD      II      RR      RR  AA      AA  CC          CC
DD      DD      II      RRRRRRRR    AA      AA  CC          CC
DD      DD      II      RRRRRRRR    AA      AA  CC          CC
DD      DD      II      RR  RR      AAAAAAAAAA  CC          CC
DD      DD      II      RR  RR      AAAAAAAAAA  CC          CC
DD      DD      II      RR    RR    AA      AA  CC          CC              ....
DD      DD      II      RR    RR    AA      AA  CC          CC              ....
DDDDDDDD      IIIIII    RR      RR  AA      AA    CCCCCCCC    CCCCCCCC      ....
DDDDDDDD      IIIIII    RR      RR  AA      AA    CCCCCCCC    CCCCCCCC      ....


LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0 MODULE DIRACC (
0               LANGUAGE (BLISS32),
0               IDENT = 'V04-000'
0               ) =
1 BEGIN
1
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1
1 !++
1 !
1 ! FACILITY:  F11ACP Structure Level 2
1 !
1 ! ABSTRACT:
1 !
1 !       This routine "accesses" a directory and readies it for processing.
1 ! ENVIRONMENT:
1 !
1 !       STARLET operating system, including privileged system services
1 !       and internal exec routines.
1 !
1 !--
1 !
1 !
1 ! AUTHOR:  Andrew C. Goldstein,  CREATION DATE:  2-Jan-1977  22:53
1 !
1 ! MODIFIED BY:
1 !
1 !       V03-015 CDS0009         Christian D. Saether    29-Aug-1984
1 !               Deal with potential multiple headers caused by
1 !               ACL's.
1 !
1 !       V03-014 CDS0008         Christian D. Saether    31-July-1984
1 !               Changes to accomodate block-cache based directory
1 !               index caching.
1 !
```

```
;    58          0058    1 !             V03-013 LMP0275         L. Mark Pilant,         12-Jul-1984  14:56
;    59          0059    1 !                     Initialize the ACL info in the ORB to be a null descriptor
;    60          0060    1 !                     list rather than an empty queue.  This avoids the overhead
;    61          0061    1 !                     of locking and unlocking the ACL mutex, only to find out
;    62          0062    1 !                     that the ACL was empty.
;    63          0063    1 !
;    64          0064    1 !             V03-012 CDS0007         Christian D. Saether    22-Apr-1984
;    65          0065    1 !                     Modify access arbitration.
;    66          0066    1 !
;    67          0067    1 !             V03-011 ACG0412         Andrew C. Goldstein,    22-Mar-1984  18:23
;    68          0068    1 !                     Implement agent access mode support; add access mode
;    69          0069    1 !                     check protection call
;    70          0070    1 !
;    71          0071    1 !             V03-010 CDS0006         Christian D. Saether    29-Dec-1983
;    72          0072    1 !                     Use L_NORM linkage and BIND_COMMON macro.
;    73          0073    1 !
;    74          0074    1 !             V03-009 CDS0005         Christian D. Saether     3-Oct-1983
;    75          0075    1 !                     Save/restore CURR_LCKINDX instead of PRIM_LCKINDX.
;    76          0076    1 !
;    77          0077    1 !             V03-008 CDS0004         Christian D. Saether    14-Sep-1983
;    78          0078    1 !                     Modify interface to SERIAL_FILE.
;    79          0079    1 !
;    80          0080    1 !             V03-007 ACG0354         Andrew C. Goldstein,    13-Sep-1983  16:16
;    81          0081    1 !                     Add alternate access validation mask
;    82          0082    1 !
;    83          0083    1 !             V03-006 CDS0003         Christian D. Saether     4-May-1983
;    84          0084    1 !                     Add directory file processing synchronization
;    85          0085    1 !                     using SERIAL_FILE.
;    86          0086    1 !                     Move directory protection check and access arbitration
;    87          0087    1 !                     checks to path that sets up DIR_FCB.
;    88          0088    1 !
;    89          0089    1 !             V03-005 CDS0002         Christian D. Saether     9-Apr-1983
;    90          0090    1 !                     Reflect changes to ACCESS_LOCK interface.
;    91          0091    1 !
;    92          0092    1 !             V03-004 CDS0001         Christian D. Saether    20-Jan-1982
;    93          0093    1 !                     Use ACCESS_LOCK routine for access arbitration.
;    94          0094    1 !
;    95          0095    1 !             V03-003 ACG0308         Andrew C. Goldstein,    14-Jan-1983  15:07
;    96          0096    1 !                     Fix consistency problems with FCB linkages
;    97          0097    1 !
;    98          0098    1 !             V03-002 LMP0059         L. Mark Pilant,         22-Dec-1982  8:22
;    99          0099    1 !                     Eliminate the call to ACL_BUILDACL as the ACL is now built by
;   100          0100    1 !                     the FCB creation/entending routines.
;   101          0101    1 !
;   102          0102    1 !             V03-001 LMP0036         L. Mark Pilant,          6-Aug-1982  10:30
;   103          0103    1 !                     Add support for ACL's.
;   104          0104    1 !
;   105          0105    1 !             V02-005 ACG0229         Andrew C. Goldstein,    23-Dec-1981  20:55
;   106          0106    1 !                     Count directory LRU hits and misses
;   107          0107    1 !
;   108          0108    1 !             V02-004 ACG33975        Andrew C. Goldstein,    13-Mar-1981  15:46
;   109          0109    1 !                     Fix losing directory FCB's when LRU is disabled
;   110          0110    1 !
;   111          0111    1 !             V02-003 ACG0167         Andrew C. Goldstein,    16-Apr-1980  19:25
;   112          0112    1 !                     Previous revision history moved to F11B.REV
;   113          0113    1 !**
;   114          0114    1
;  115          0115  1
;  116          0116  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;  117          0117  1 REQUIRE 'SRC$:FCPDEF.B32';
;  118          1108  1
;  120        1109  1  GLOBAL ROUTINE DIR_ACCESS (FIB, WRITE) : L_NORM NOVALUE =
;  121        1110  1
;  122        1111  1  !++
;  123        1112  1  !
;  124        1113  1  ! FUNCTIONAL DESCRIPTION:
;  125        1114  1  !
;  126        1115  1  !       This routine "accesses" a directory and readies it for processing.
;  127        1116  1  !
;  128        1117  1  ! CALLING SEQUENCE:
;  129        1118  1  !       DIR_ACCESS (ARG1, ARG2)
;  130        1119  1  !
;  131        1120  1  ! INPUT PARAMETERS:
;  132        1121  1  !       ARG1: address of FIB
;  133        1122  1  !       ARG2: 0 if read only
;  134        1123  1  !             1 if read/write
;  135        1124  1  !
;  136        1125  1  ! IMPLICIT INPUTS:
;  137        1126  1  !       NONE
;  138        1127  1  !
;  139        1128  1  ! OUTPUT PARAMETERS:
;  140        1129  1  !       NONE
;  141        1130  1  !
;  142        1131  1  ! IMPLICIT OUTPUTS:
;  143        1132  1  !       DIR_FCB: address of FCB for directory
;  144        1133  1  !
;  145        1134  1  ! ROUTINE VALUE:
;  146        1135  1  !       NONE
;  147        1136  1  !
;  148        1137  1  ! SIDE EFFECTS:
;  149        1138  1  !       FCB may be created
;  150        1139  1  !       directory LRU may be altered
;  151        1140  1  !       directory file header may be read
;  152        1141  1  !
;  153        1142  1  !--
;  154        1143  1
;  155        1144  2  BEGIN
;  156        1145  2
;  157        1146  2  MAP
;  158        1147  2          FIB             : REF BBLOCK;   ! FIB argument
;  159        1148  2
;  160        1149  2  LOCAL
;  161        1150  2          CURR_LKMODE,                    ! current access lock mode
;  162        1151  2          SAV_LCKINDX,
;  163        1152  2          STATUS,                         ! protection check status value
;  164        1153  2          FCB             : REF BBLOCK,   ! FCB address
;  165        1154  2          HEADER          : REF BBLOCK;   ! directory header address
;  166        1155  2
;  167        1156  2  BIND_COMMON;
;  168        1157  2
;  169        1158  2  EXTERNAL ROUTINE
;  170        1159  2          ARBITRATE_ACCESS : L_JSB_2ARGS, ! access arbitration
;  171        1160  2          BUILD_EXT_FCBS  : L_NORM NOVALUE, ! build extension headers
;  172        1161  2          CONV_ACCLOCK    : L_NORM,       ! convert access lock
;  173        1162  2          CREATE_FCB      : L_NORM,       ! create fcb
;  174        1163  2          MAKE_DIRINDX    : L_NORM,       ! validate/create directory index
;  175        1164  2          REBLD_PRIM_FCB  : L_NORM NOVALUE, ! refresh fcb from header
;  176        1165  2          SERIAL_FILE     : L_NORM,       ! interlock directory processing.
```

```
2               SWITCH_VOLUME   : L_NORM,       ! switch to correct volume
2               SEARCH_FCB      : L_NORM,       ! search FCB list
2               READ_HEADER     : L_NORM,       ! read file header
2               CHECK_PROTECT   : L_NORM;       ! check file protection
2
2
2   ! Switch context to the volume of the specified RVN.
2   !
2
2   SWITCH_VOLUME (.FIB[FIB$W_DID_RVN]);
2
2   ! First find the FCB of the directory, if any exists.
2   !
2
2   IF (FCB = .DIR_FCB) EQL 0
2   THEN
3       BEGIN
3       LOCAL
3           FCBVALID : INITIAL (0);
3
3   ! Synchronize further file processing on this directory.
3   !
3
3       SAV_LCKINDX = .CURR_LCKINDX;
3
3       DIR_LCKINDX = SERIAL_FILE (FIB [FIB$W_DID]);
3
3       FCB = SEARCH_FCB (FIB[FIB$W_DID]);
3
3       IF .FCB NEQ 0
3       THEN
3           FCBVALID = MAKE_DIRINDX (.FCB);
3
3       IF NOT .FCBVALID
3       THEN
4           BEGIN
4
4   ! This flag tells READ_HEADER to not update the FILE_HEADER cell.
4   !
4
4           STSFLGS [STS_LEAVE_FILEHDR] = 1;
4
4           HEADER = READ_HEADER (FIB [FIB$W_DID], .FCB);
4
4           IF .FCB EQL 0
4           THEN
5               BEGIN
5               FCB = CREATE_FCB (.HEADER);
5               MAKE_DIRINDX (.FCB);
5               END
4           ELSE
4               REBLD_PRIM_FCB (.FCB, .HEADER);
4
4           IF .HEADER [FH2$W_EX_FIDNUM] NEQ 0
4               OR .HEADER [FH2$B_EX_FIDNMX] NEQ 0
4           THEN
4               BUILD_EXT_FCBS (.HEADER, .FCB);
```

```
;   234          1223  4
;   235          1224  4   ! Check the file attributes to make sure that
;   236          1225  4   ! this is a real directory file.
;   237          1226  4   !
;   238          1227  4
;   239          1228  4           IF NOT .HEADER[FH2$V_DIRECTORY]
;   240          1229  4               OR NOT .HEADER[FH2$V_CONTIG]
;   241          1230  4               OR .BBLOCK [HEADER[FH2$W_RECATTR], FAT$B_RTYPE] NEQ FAT$C_VARIABLE
;   242          1231  4               OR .BBLOCK [HEADER[FH2$W_RECATTR], FAT$B_RATTRIB] NEQ FAT$M_NOSPAN
;   243          1232  4           THEN ERR_EXIT (SS$_BADIRECTORY);
;   244          1233  4
;   245          1234  4           FCB [FCB$W_DIRSEQ] = .FCB [FCB$W_DIRSEQ] + 1;
;   246          1235  4
;   247          1236  3           END;
;   248          1237  3
;   249          1238  3       CURR_LCKINDX = .SAV_LCKINDX;
;   250          1239  3
;   251          1240  3       IF .FCB [FCB$W_FID_SEQ] NEQ .FIB [FIB$W_DID_SEQ]
;   252          1241  3       THEN
;   253          1242  3           ERR_EXIT (SS$_FILESEQCHK);
;   254          1243  3
;   255          1244  3       DIR_FCB = .FCB;                                 ! store global FCB address
;   256          1245  3
;   257          1246  3   ! Check directory file protection. For alternate access validation,
;   258          1247  3   ! delete access implies write to the directory.
;   259          1248  3   !
;   260          1249  3
;   261          1250  3       STATUS = CHECK_PROTECT (.WRITE, .HEADER, .FCB, 0,
;   262          1251  4                       (IF .BBLOCK [FIB[FIB$L_ALT_ACCESS], ARM$V_DELETE]
;   263          1252  3                        THEN ARM$M_WRITE ELSE 0),
;   264          1253  3                       .FIB[FIB$V_ALT_REQ]);
;   265          1254  3
;   266          1255  3       IF .STATUS EQL SS$_NOTALLPRIV
;   267          1256  3       THEN FIB[FIB$V_ALT_GRANTED] = 0;
;   268          1257  3
;   269          1258  3   ! Arbitrate the access interlocks.
;   270          1259  3   !
;   271          1260  3
;   272          1261  3       CURR_LKMODE = .FCB [FCB$B_ACCLKMODE];
;   273          1262  3
;   274          1263  3       IF NOT ARBITRATE_ACCESS (IF .WRITE THEN FIB$M_WRITE ELSE 0, .FCB)
;   275          1264  3       THEN ERR_EXIT (SS$_ACCONFLICT);
;   276          1265  3
;   277          1266  3       CONV_ACCLOCK (.CURR_LKMODE, .FCB);
;   278          1267  3
;   279          1268  2       END;
;   280          1269  2
;   281          1270  2   RETURN 1;
;   282          1271  2
;   283          1272  1   END;                                    ! end of routine DIR_ACCESS


                                                             .TITLE  DIRACC
                                                             .IDENT  \V04-000\

                                                             .EXTRN  ARBITRATE_ACCESS
                                                             .EXTRN  BUILD_EXT_FCBS, CONV_ACCLOCK
```

```
                                          .EXTRN  CREATE_FCB, MAKE_DIRINDX
                                          .EXTRN  REBLD_PRIM_FCB, SERIAL_FILE
                                          .EXTRN  SWITCH_VOLUME, SEARCH_FCB
                                          .EXTRN  READ_HEADER, CHECK_PROTECT

                                          .PSECT  $CODE$,NOWRT,2

                         001C 00000       .ENTRY  DIR_ACCESS, Save R2,R3,R4       ; 1109
           50      04  AC  D0 00002       MOVL    FIB, R0                         ; 1175
           7E      0E  A0  3C 00006       MOVZWL  14(R0), -(SP)                   ;
     0000G CF          01  FB 0000A       CALLS   #1, SWITCH_VOLUME               ;
           53    00D0  CA  D0 0000F       MOVL    208(BASE), FCB                  ; 1180
                       01  13 00014       BEQL    1$                              ;
                           04 00016       RET                                     ;
                       52  D4 00017 1$:   CLRL    FCBVALID                        ; 1182
           54      14  AA  D0 00019       MOVL    20(BASE), SAV_LCKINDX           ; 1189
7E      04 AC          0A  C1 0001D       ADDL3   #10, FIB, -(SP)                 ; 1191
     0000G CF          01  FB 00022       CALLS   #1, SERIAL_FILE                 ;
     00D4  CA          50  D0 00027       MOVL    R0, 212(BASE)                   ;
7E      04 AC          0A  C1 0002C       ADDL3   #10, FIB, -(SP)                 ; 1193
     0000G CF          01  FB 00031       CALLS   #1, SEARCH_FCB                  ;
           53          50  D0 00036       MOVL    R0, FCB                         ;
                       0A  13 00039       BEQL    2$                              ; 1195
                       53  DD 0003B       PUSHL   FCB                             ; 1197
     0000G CF          01  FB 0003D       CALLS   #1, MAKE_DIRINDX                ;
           52          50  D0 00042       MOVL    R0, FCBVALID                    ;
           62          52  E8 00045 2$:   BLBS    FCBVALID, 9$                    ; 1199
        A6 AA          08  88 00048       BISB2   #8, -90(BASE)                   ; 1206
                       53  DD 0004C       PUSHL   FCB                             ; 1208
7E      04 AC          0A  C1 0004E       ADDL3   #10, FIB, -(SP)                 ;
     0000G CF          02  FB 00053       CALLS   #2, READ_HEADER                 ;
           52          50  D0 00058       MOVL    R0, HEADER                      ;
                       53  D5 0005B       TSTL    FCB                             ; 1210
                       13  12 0005D       BNEQ    3$                              ;
                       52  DD 0005F       PUSHL   HEADER                          ; 1213
     0000G CF          01  FB 00061       CALLS   #1, CREATE_FCB                  ;
           53          50  D0 00066       MOVL    R0, FCB                         ;
                       53  DD 00069       PUSHL   FCB                             ; 1214
     0000G CF          01  FB 0006B       CALLS   #1, MAKE_DIRINDX                ;
                       09  11 00070       BRB     4$                              ; 1210
                       52  DD 00072 3$:   PUSHL   HEADER                          ; 1217
                       53  DD 00074       PUSHL   FCB                             ;
     0000G CF          02  FB 00076       CALLS   #2, REBLD_PRIM_FCB              ;
                   0E  A2  B5 0007B 4$:   TSTW    14(HEADER)                      ; 1219
                       05  12 0007E       BNEQ    5$                              ;
                   13  A2  95 00080       TSTB    19(HEADER)                      ; 1220
                       07  13 00083       BEQL    6$                              ;
                       0C  BB 00085 5$:   PUSHR   #^M<R2,R3>                      ; 1222
     0000G CF          02  FB 00087       CALLS   #2, BUILD_EXT_FCBS              ;
11      35 A2          05  E1 0008C 6$:   BBC     #5, 53(HEADER), 7$              ; 1228
                   34  A2  95 00091       TSTB    52(HEADER)                      ; 1229
                       0C  18 00094       BGEQ    7$                              ;
           02      14  A2  91 00096       CMPB    20(HEADER), #2                  ; 1230
                       06  12 0009A       BNEQ    7$                              ;
           08      15  A2  91 0009C       CMPB    21(HEADER), #8                  ; 1231
                       05  13 000A0       BEQL    8$                              ;
                 0828  8F  BF 000A2 7$:   CHMU    #2088                           ; 1232
```

```
                                                         04 000A6          RET                                                 ;
                                                 42   A3 B6 000A7 8$:      INCW    66(FCB)                                     ; 1234
                                  14    AA            54 D0 000AA 9$:      MOVL    SAV_LCKINDX, 20(BASE)                       ; 1238
                                        50       04   AC D0 000AE          MOVL    FIB, R0                                     ; 1240
                                  0C    A0       26   A3 B1 000B2          CMPW    38(FCB), 12(R0)                             ;
                                                      05 13 000B7          BEQL    10$                                         ;
                                               08B8   8F BF 000B9          CHMU    #2232                                       ; 1242
                                                         04 000BD          RET                                                 ;
                                00D0    CA            53 D0 000BE 10$:     MOVL    FCB, 208(BASE)                              ; 1244
                                        50       04   AC D0 000C3          MOVL    FIB, R0                                     ; 1253
           7E       38   A0             01            00 EF 000C7          EXTZV   #0, #1, 56(R0), -(SP)                       ;
                         04       3C    A0            03 E1 000CD          BBC     #3, 60(R0), 11$                             ; 1251
                                                      02 DD 000D2          PUSHL   #2                                          ;
                                                      02 11 000D4          BRB     12$                                         ;
                                                      7E D4 000D6 11$:     CLRL    -(SP)                                       ;
                                                      7E D4 000D8 12$:     CLRL    -(SP)                                       ; 1250
                                                      0C BB 000DA          PUSHR   #^M<R2,R3>                                  ;
                                                 08   AC DD 000DC          PUSHL   WRITE                                       ;
                               0000G    CF            06 FB 000DF          CALLS   #6, CHECK_PROTECT                           ;
                            00000681    8F            50 D1 000E4          CMPL    STATUS, #1665                               ; 1255
                                                      08 12 000EB          BNEQ    13$                                         ;
                                        50       04   AC D0 000ED          MOVL    FIB, R0                                     ; 1256
                                  38    A0            02 8A 000F1          BICB2   #2, 56(R0)                                  ;
                                        52       0B   A3 9A 000F5 13$:     MOVZBL  11(FCB), CURR_LKMODE                        ; 1261
                                        07       08   AC E9 000F9          BLBC    WRITE, 14$                                  ; 1263
                                        50     0100   8F 3C 000FD          MOVZWL  #256, R0                                    ;
                                                      02 11 00102          BRB     15$                                         ;
                                                      50 D4 00104 14$:     CLRL    R0                                          ;
                                        51            53 D0 00106 15$:     MOVL    FCB, R1                                     ;
                                                   0000G 30 00109          BSBW    ARBITRATE_ACCESS                            ;
                                        05            50 E8 0010C          BLBS    R0, 16$                                     ;
                                               0800   8F BF 0010F          CHMU    #2048                                       ; 1264
                                                         04 00113          RET                                                 ;
                                                      0C BB 00114 16$:     PUSHR   #^M<R2,R3>                                  ; 1266
                               0000G    CF            02 FB 00116          CALLS   #2, CONV_ACCLOCK                            ;
                                                         04 0011B          RET                                                 ; 1272

; Routine Size:  284 bytes,    Routine Base:  $CODE$ + 0000


;  284           1273  1
;  285           1274  1 END
;  286           1275  0 ELUDOM




;                                       PSECT SUMMARY
;
;       Name                          Bytes                          Attributes
;
;   $CODE$                                 284  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;                                Library Statistics
;
;                                               -------- Symbols --------      Pages       Processing
;          File                                 Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1                18619        42         0      1000          00:01.8







;                                       COMMAND QUALIFIERS

;          BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:DIRACC/OBJ=OBJ$:DIRACC MSRC$:DIRACC/UPDATE=(ENH$:DIRACC)

; Size:          284 code + 0 data bytes
; Run Time:         00:18.8
; Elapsed Time:     00:35.7
; Lines/CPU Min:      4080
; Lexemes/CPU-Min: 47273
; Memory Used:  242 pages
; Compilation Complete
```

0169 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY